AMENDMENTS TO THE CLAIMS:

1. (Currently amended) A computer, comprising:

a processor;

a memory system; and

a co-processing ~~unit; and~~ unit with an associated ~~a~~ plurality of data registers for data exchange ~~with said co-processing unit~~,

wherein said computer is controlled to implement a method of increasing efficiency in executing a matrix operation that uses matrix data in a standard format, said standard format comprising one of a column major format and a row major format, said matrix operation being executed in said co-processing unit, said method comprising:

for matrix data stored in said standard format in said memory system, wherein said matrix data comprises data of any of a complete matrix, a complete submatrix, or a part of a matrix or submatrix, using said processor to separate said matrix data into blocks of data, each said block having a size p-by-q; and

rearranging by said processor and placing in said memory system of said computer, for retrieval in a repetitive manner for executing said matrix operation, said blocks of data to be contiguous ~~blocks of contiguous~~ data, wherein data within said blocks retain an original matrix data content but said blocks are moved to be in an ordering different from an original ordering of said blocks within said matrix, such that said matrix data is represented in a ~~nonstandard~~ format that permits said matrix data to be moved from said memory system into a position in said plurality of data registers for performing said matrix operation more quickly than if said matrix data had been moved as stored in said standard format.

2. (Currently amended) The computer of claim 22, wherein said co-processing unit comprises a floating point unit (FPU) and said loading of said matrix data into said set of data registers comprises loading said blocks from said storage into a subset of data registers in said set of data registers, using a deviation from a normal slow floating point loading instruction of the floating point unit (FPU) of the computer by loading data words in a new word order thereby producing a fast optimal loading of said data.

3. (Canceled)

4. (Previously presented) The computer of claim 1, wherein said size p-by-q comprises a 2-by-2 block.

5. (Currently amended) The computer of claim 2, wherein said deviation from normal floating point loading comprises a crisscrossing or achieves an effect of a crisscrossing of elements about a diagonal of said blocks.

6. (Currently amended) The computer of claim 2, said method further comprising:

selectively, at least one of loading input data and storing a result of said matrix operation into or out of said co-processing unit from an L1 cache or said memory by at least one of a subset of optimal load and store instructions, said loading and storing being dictated by an optimal FPU loading or storage instruction.

7. (Previously presented) The computer of claim 2, wherein said deviation of said normal floating point loading instruction, in combination with said nonstandard format, provides a result data of a transpose of said matrix data to reside in said data registers of said FPU.

8. (Currently amended) The computer of claim 2, wherein said loading comprises a crisscrossing or achieves an effect of a 2 x 2 crisscrossing technique.

9. (Currently amended) The computer of claim 6, wherein said linear algebra operation comprises one of a BLAS kernel or of a factorization kernel.

10-16. (Canceled)

17. (Currently amended) A computer-readable storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of storing information of a matrix in a register block data format, said method comprising:

receiving data for a matrix, said data comprising one of a complete matrix data, a complete submatrix data, and a partial matrix or submatrix data, said matrix data being stored in one of a standard column format and a standard row format;

dividing said matrix data into blocks, each said block having a size p-by-q; and

~~at least one of:~~

~~storing elements in at least one of said blocks in at least one of a cache and a memory in a format in which is elements of said block occupy a location different from an original location in said block;~~

storing, for a repetitive retrieval, said blocks of size p-by-q in a memory in a format in which ~~at least one~~ said ~~block occupies~~ blocks occupy a position different from its original position in said matrix while maintaining an original data content within each said block, said register data block format thereby representing the matrix data in a format that is no longer ~~be~~ in either of said standard column format or said standard row format.

18. (Currently amended) The computer-readable storage medium of claim 17, said method further comprising:

repetitively loading said blocks from said memory into a plurality of data registers so that a format of data in said data registers comprises or results in an effect of a transpose data of said matrix.

19. (Currently amended) The computer-readable storage medium of claim 18, wherein said repetitively loading comprises a crisscrossing or achieves an effect of a loading using a 2 x 2 crisscrossing technique.

20. (Canceled)

21. (Currently amended) The computer of claim 1, said method further comprising:

repetitively retrieving said matrix data from said memory system as matrix data in ~~in said~~ a new format; and

loading said matrix data into at least a subset of said set of data registers in a new or optimal format, said optimal format comprising a format of said matrix data in said data registers such that a minimal possible time is required to utilize said matrix data in said data registers in said matrix operation in said co-processing unit.

22. (Currently amended) The computer of claim 21, wherein said computer includes at least one of a machine architecture and an instruction set having one or more features that are less than optimal for executing said matrix operation in said standard format with said co-processing unit, and said new format of matrix data and said fast loading into said data

registers together provide a mechanism that overcomes said one or more features that are less than optimal for executing said matrix operation.

23. (Currently amended) A computer comprising:

a processor;

a storage; and

a co-processing unit,

said computer configured to implement a method of increasing efficiency in executing a matrix operation that uses matrix data in a standard format, said standard format comprising one of a column major format and a row major format, said method comprising:

converting, by said processor, at least a part of said matrix data into a new or optimal matrix format comprising contiguous data that no longer represents said matrix data in said standard format, said optimal matrix format comprising a representation of a subset of said matrix data that is predetermined to permit a loading of said matrix data from said storage into said co-processing unit optimally to perform said matrix operation such that a minimal possible time in said processing unit is used to utilize said matrix data of said matrix operation, said optimal matrix format comprising a re-arrangement of blocks of said matrix data wherein data within each block retains its original values.

24. (Previously presented) The computer of claim 23, said method further comprising repetitively loading a selected block of matrix data in said optimal matrix format into said processing unit for executing said matrix operation, wherein said loading comprises repetitively placing data of said selected block into predetermined registers of a register set of said co-processing unit.

25. (Currently amended) The computer of claim 24, said method further comprising:

processing, by said co-processing unit, said matrix operation on data in said selected block, a result of said processing being stored in predetermined registers of said register set; and

storing said result from said predetermined registers of said register set into said storage ~~in a format corresponding to said optimal matrix format~~.

26. (Currently amended) A computer, comprising:

a processor;

a storage; and

a co-processing ~~unit; and~~ with an associated ~~a~~ plurality of data registers for data exchange ~~with said co-processing unit~~,

said computer having at least one of a machine architecture and an instruction set having one or more features that are less than optimal for executing a matrix operation, thereby causing a disadvantage in processing data for said matrix operation, said computer configured to implement a method of overcoming said disadvantage by software instructions, said method comprising:

rearranging, by said processor, at least a part of matrix data to be used in said matrix operation into a plurality of blocks, each block having size p-by-q, such that said matrix data is no longer stored in a standard matrix format comprising one of a row major format and a column major format, said rearranged matrix data in said blocks being stored in said storage as contiguous blocks of contiguous data in a new format such that an original content of data within said blocks is retained but an ordering of said blocks is changed,

wherein said new format of said matrix data is predetermined to allow said matrix data to be placed from said storage into said co-processing unit for processing said

matrix data in said matrix operation such that said disadvantage on said computer is overcome and said matrix processing will be correctly executed.

27. (Currently amended) The computer of claim 26, said method further comprising:

repetitively loading said matrix data in said new format from said storage into at least a subset of said data registers of said co-processing unit in a new or optimal format that allows a minimal possible time in said processing unit to utilize said matrix data in said matrix operation.

28. (Currently amended) A computer, comprising:

a processor;

a storage; and

a co-processing ~~unit; and~~ unit with an associated ~~a~~ plurality of data registers for data exchange ~~with said co-processing unit,~~

said computer configured to implement a method of overcoming a hardware disadvantage on said computer relative to a specific processing on a specific computer architecture/set of instructions using said co-processing unit, said hardware disadvantage reducing an efficiency of said specific processing, said method comprising:

using first software instructions to preliminarily process input data by said processor in a manner to generate a first error relative to said specific processing, said first error comprising a conversion of said input data into a predetermined new format of said input data; and

using second software instructions to subsequently process said input data in said new format in a manner to generate a correcting error relative to said specific processing,

said correcting error comprising a loading said input data into said plurality of data registers in a new word order of said input data,

wherein first software instructions in combination with said second software instructions overcome said disadvantage and computes a correct result.

29. (Currently amended) The computer of claim 28, wherein said specific processing comprises a matrix operation, said disadvantage comprises a loading of matrix data from said storage into said co-processing unit that causes a non optimal processing of said matrix data in said matrix operation, said first error comprises storing said matrix data in said storage in a format that converts said matrix data from a standard column major or row major format into a new format predetermined to overcome said disadvantage when said data is subjected to said correcting error such that an original content of data within said blocks is retained but an ordering of said blocks is changed, and said correcting error comprises loading said data in said new format from said storage into said plurality of data registers using a loading format comprising a non standard word order of said matrix data, permitting said loading to be done optimally and said matrix processing to be done correctly.